Name : Krish Singh

Roll: AID25072

Branch: B.Tech AI & DS 2nd sem

LAB EXCERCISE 2

1. Row vectors of matrix product AB is a linear combination of row vectors of B. Thus row space of AB and B are same (rref(AB) will be same as rref(B)). Using this concept, we can obtain infinitely many matrices that has the same rref as B, by pre-multiplying it by any suitable matrix A.

a. Create 5 different 3×4 matrices that have the rref matrix as

R= [1 0 1 0 ;

1 1 0 0;

0 0 0 1].

```
R=[1 0 1 0; 1 1 0 0; 0 0 0 1];
for i = 1:5
    A=randi([0,9],3,3)*R
    rref(A)
end
```

```
A = 3×4
    14     5     9     8
     8     1     7     8
    12     9     3     2
ans = 3×4
     1     0     1     0
     0     1    -1     0
     0     0     0     1
A = 3×4
     8     3     5     4
     1     1     0     0
     5     1     4     5
ans = 3×4
     1     0     1     0
     0     1    -1     0
     0     0     0     1
A = 3×4
    10     6     4     3
     6     0     6     5
     6     0     6     6
ans = 3×4
     1     0     1     0
     0     1    -1     0
     0     0     0     1
A = 3×4
    13     9     4     1
    13     5     8     6
    10     3     7     7
ans = 3×4
```

```
     1     0     1     0
     0     1    -1     0
     0     0     0     1
A = 3×4
     5     1     4     5
     2     2     0     4
     6     4     2     8
ans = 3×4
     1     0     1     0
     0     1    -1     0
     0     0     0     1
```

b. Create a 5×4 matrix that has non-zero rows in rref same as in

R=[1 0 1 0 ;

1 1 0 0 ;

0 0 0 1]

```
B=randi([0,9],5,3)*R
```

```
B = 5×4
    11     6     5     2
    11     2     9     2
    12     6     6     6
    15     6     9     8
     2     0     2     3
```

```
rref(B)
```

```
ans = 5×4
     1     0     1     0
     0     1    -1     0
     0     0     0     1
     0     0     0     0
     0     0     0     0
```

c. Create a 4×4 matrix that has non-zero rows in rref same as in :

R=[1 0 1 0 ;

1 1 0 0 ;

0 0 0 1]

```
B = randi([0,9],4,3)*R
```

```
B = 4×4
    10     3     7     4
    15     9     6     4
     0     0     0     7
    10     4     6     3
```

```
rref(B)
```

```
ans = 4×4
     1     0     1     0
```

```
     0     1    -1     0
     0     0     0     1
     0     0     0     0
```

2. Every matrix Am×n matrix of rank r can be decomposed into A=CR, where C is an m×r matrix with column vectors as the independent columns of A and R is an r×n matrix with row vectors as non-zero rows in rref(A).

C and R can be obtained in MATLAB using these lines of code:

%Enter matrix A [RR, ic]=rref(A);

r = length(ic);

R=RR(1:r, :);

C=A(:, ic);

%Verify by checking A=CR Decompose the following matrices in A=CR form and verify the result.

a. A=(1 3 4 7; 2 4 6 10; 3 5 8 13; 4 6 10 16)

b. B=(35 28 35 14; 40 32 32 14 ;40 32 24 12; 75 60 67 28)

```
A=[ 1 3 4 7; 2 4 6 10; 3 5 8 13; 4 6 10 16 ];
[RR,ic]=rref(A);
r=length(ic);
R=RR(1:r,:);
C=A(:,ic);
if(A==C*R)
    disp("Results Verified i.e A=CR")
else
    disp("Results not verified i.e A!=CR")
end
```

```
Results Verified i.e A=CR
```

```
B=[35 28 35 14; 40 32 32 14 ;40 32 24 12; 75 60 67 28];
[RR,ic]=rref(B);
r=length(ic);
R=RR(1:r,:);
C=RR(:,ic);
if(B==C*R)
    disp("Results verified i.e C=R")
else
    disp("Results not verified i.e C!=R")
end
```

```
Results not verified i.e C!=R
```

c. Any random 7×9 integer matrix with rank 3.

```
A=randi([0,9],7,3)*randi([0,9],3,9)
```

```
A = 7×9
    88   123    49    53    95    67    94    92    83
    58    69    52    44    59    52    82    59    47
    18    24    11    14    20    14    12    15     8
    46    48    51    43    46    45    64    41    22
    70    99    38    39    75    53    82    77    75
   106   132    83    82   112    90   116    99    68
    46    63    26    33    51    35    34    41    27
```

```
rank(A)
```

```
ans = 3
```

```
[RR,ic]=rref(A);
r=length(ic);
R=RR(1:r,:);
C=RR(:,ic);
if(A==C*R)
    disp("Results verified i.e A=CR")
else
    disp("Results not verified i.e A!=CR")
end
```

```
Results not verified i.e A!=CR
```

d. Any random 9×4 integer matrix with rank 3.

```
B=randi([0,9],9,3)*randi([0,9],3,4)
```

```
B = 9×4
    86    61    53    40
   138    99   126    66
    96    72    73    30
    97    73    60    27
   107    81    74    29
    33    26    38     7
    58    42    65    28
    93    68    73    37
    50    37    64    22
```

```
rank(B)
```

```
ans = 3
```

```
[RR,ic]=rref(B);
r=length(ic);
R=RR(1:r,:);
C=RR(:,ic);
if(B==C*R)
    disp("Results verified i.e A=CR")
else
    disp("Results not verified i.e A!=CR")
end
```

```
Results not verified i.e A!=CR
```

3. Left Inverse of a matrix: If all columns of matrix A of size mxn are independent (or rank(A) equals n), then the matrix has a left inverse such that

If all columns are independent, you may notice that 'm' should be greater than or equal to 'n'. If m = n , then it is a square matrix and left inverse is same as original inverse. Perform the following experiments:

a. Create a random 4 by 3 matrix

```
A=randi([0,9],4,3)
```

```
A = 4×3
     0     6     6
     7     3     7
     2     7     4
     4     3     0
```

b. Check whether rank (A) = n =3

```
n=rank(A)
```

```
n = 3
```

c. Find Left inverse of A

```
ALI=inv(A'*A)*A'
```

```
ALI = 3×4
   -0.0912    0.0804   -0.0040    0.1112
    0.0274   -0.0894    0.1154    0.0988
    0.0878    0.0991   -0.0553   -0.1459
```

d. Check whether ALI*A produces an identity matrix.

```
check=ALI*A
```

```
check = 3×3
    1.0000   -0.0000   -0.0000
   -0.0000    1.0000    0.0000
         0   -0.0000    1.0000
```

```
%can check identity matrix using order and trace comparision
tol=1e-10
```

```
tol = 1.0000e-10
```

```
if norm(check-eye(n))<tol
    disp("Matrix is an identity matrix")
else
```

```
    disp("Matrix is not an identity matrix")
end
```

```
Matrix is an identity matrix
```

4. Right Inverse of a matrix: If all rows of matrix A of size mxn are independent (or rank(A) equals m)

Create such matrices and verify the result. If all rows are independent, you may notice that 'm' should be less than or equal to 'n'. If m = n, then it is a square matrix and right inverse is same as original inverse. Perform the following experiments:

a. Create a random 5 by 7 matrix

```
A=randi([0,9],5,7)
```

```
A = 5×7
     3     4     8     9     7     8     7
     4     8     7     3     1     5     8
     2     3     3     6     8     1     7
     1     7     2     4     9     1     3
     8     3     7     8     5     4     5
```

b. Check whether rank (A) = m = 5

```
m=rank(A)
```

```
m = 5
```

c. Find right inverse of A

```
ARI=A'*(inv(A*A'))
```

```
ARI = 7×5
   -0.1010    0.0056   -0.0215    0.0054    0.1458
   -0.0367    0.0670   -0.0809    0.1117   -0.0070
    0.0308    0.0165   -0.0410   -0.0107    0.0166
    0.0377   -0.0501    0.0110   -0.0141    0.0308
    0.0136   -0.0572    0.0194    0.0691   -0.0020
    0.1263   -0.0088   -0.1009    0.0068   -0.0452
   -0.0346    0.0722    0.1840   -0.1127   -0.0635
```

d. Check whether A* ARI produces an identity matrix.

```
check=A*ARI
```

```
check = 5×5
    1.0000    0.0000   -0.0000    0.0000   -0.0000
```

```
   -0.0000    1.0000   -0.0000         0         0
   -0.0000    0.0000    1.0000    0.0000   -0.0000
   -0.0000    0.0000   -0.0000    1.0000   -0.0000
   -0.0000    0.0000   -0.0000   -0.0000    1.0000
```

```
if norm(check-eye(m))<1e-10
    disp("The matrix is an identity matrix")
else
    disp("The matrix is not a identity matrix")
end
```

```
The matrix is an identity matrix
```

5. Pseudoinverse of an m×n matrix, A

a. Generate a 9×9 random integer matrix and verify that inv(A) = pinv(A).

```
A=randi([0,9],9,9)
```

```
A = 9×9
     3     4     3     7     6     8     2     4     4
     6     3     6     1     8     7     0     2     3
     2     5     9     6     5     1     9     7     8
     5     7     9     4     7     5     6     2     3
     8     4     4     2     1     3     9     0     8
     5     4     2     0     9     5     1     7     7
     3     1     7     8     5     3     9     6     8
     2     0     7     1     6     4     7     7     5
     4     2     7     1     0     1     5     6     6
```

```
if norm(inv(a)-pinv(a))<1e-10
    disp("Pseudo inverse and inverse are equal")
else
    disp("pseudoinverse and inverse are not equal")
end
```

```
Pseudo inverse and inverse are equal
```

b. Generate a 5×4 random integer matrix and verify that pinv(A) is same as the left inverse of A.

```
A=randi([0,9],5,4)
```

```
A = 5×4
     9     3     3     7
     4     9     1     3
     0     2     0     8
     8     6     4     7
     6     6     1     5
```

```
a=pinv(A);
ALI=inv(A'*A)*A';
if norm(a-ALI)<1e-10
    disp("pseudoinverse and left inverse of A are same")
```

```
else
    disp("pseudoinverse and left inverse of A are not the same")
end
```

```
pseudoinverse and left inverse of A are same
```

c. Generate a 3×7 random integer matrix and verify that pinv(A) is same as the right inverse of A.

```
B=randi([0,9],3,7)
```

```
B = 3×7
     8     2     1     3     9     7     2
     6     1     6     4     6     3     2
     1     0     9     9     9     6     6
```

```
a=pinv(B)
```

```
a = 7×3
    0.0101    0.1403   -0.0825
    0.0137    0.0108   -0.0139
   -0.0975    0.1484   -0.0008
   -0.0125   -0.0159    0.0446
    0.0523   -0.0418    0.0227
    0.0774   -0.1100    0.0378
    0.0075   -0.0495    0.0429
```

```
BRI=B'*inv(B*B')
```

```
BRI = 7×3
    0.0101    0.1403   -0.0825
    0.0137    0.0108   -0.0139
   -0.0975    0.1484   -0.0008
   -0.0125   -0.0159    0.0446
    0.0523   -0.0418    0.0227
    0.0774   -0.1100    0.0378
    0.0075   -0.0495    0.0429
```

```
if norm(a-BRI)<1e-10
    disp("pseudoinverse and right inverse are same")
else
    disp("pseudoinverse and right inverse are not same")
end
```

```
pseudoinverse and right inverse are same
```

6. Projection of a vector v on a subspace S, which is a column space of matrix A(with all independent columns) is p = A(ATA)-1ATv. If some columns in A are dependent (i.e., if Rank(Am×n) < n), then the projection of a vector v on a subspace S, which is a column space of matrix A is p = A*pinv(A)*v, where pinv(A) is the pseudo inverse of A.

a. Find the projection of v=(-3 -3 8 9) on the space spanned by (3 1 0 1),(1 2 1 1),(-1 0 2 -1).

```
v=[-3;-3;8;9];
A=[3 1 -1;
    1 2 0;
    0 1 2;
    1 1 -1];
rank(A)
```

```
ans = 
3
```

```
if rank(A)<size(A,2)
    disp("Some columns are dependent")
    p=A*pinv(A)*v
else
    disp("All columns are independent")
    p=A*inv(A'*A)*A'*v
end
```

```
All columns are independent
p = 4×1
   -2.0000
    3.0000
    4.0000
   -0.0000
```

b. Find the projection of u=(2 1 3) on the space spanned by (1 1 0),(1 2 1).

```
u=[2;1;3]
```

```
u = 3×1
     2
     1
     3
```

```
A=[1 1;
    1 2;
    0 1];
rank(A)
```

```
ans = 
2
```

```
if rank(A)<size(A,2)
    disp("Some columns are dependent")
    p=A*pinv(A)*u
else
    disp("All columns are independent")
    p=A*inv(A'*A)*A'*u
end
```

```
All columns are independent
p = 3×1
    0.6667
    2.3333
    1.6667
```

c. Find the projection of u=(2 1 3) on the column space of =(1 1 2 4; 1 2 3 6; 0 1 1 2).

```
u=[2;1;3]
```

```
u = 3×1
     2
     1
     3
```

```
A=[1 1 2 4;
    1 2 3 6;
    0 1 1 2];
rank(A)
```

```
ans =
2
```

```
if rank(A)<size(A,2)
    disp("Some columns are dependent")
    p=A * pinv(A) * u
else
    disp("All columns are independent")
    p=A*inv(A'*A)*A'*u
end
```

```
Some columns are dependent
p = 3×1
    0.6667
    2.3333
    1.6667
```